# CHROOT-01

Unset root SUID after calling chroot()

Sean Barnum, Cigital, Inc. [vita[1]]

2007-03-19

# Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 4389 bytes

<table>
<tr><td><b>Attack Category</b></td><td>- Privilege Exploitation</td></tr>
<tr><td><b>Vulnerability Category</b></td><td>- Privilege escalation problem</td></tr>
<tr><td><b>Software Context</b></td><td>- Process Management<br>- Authorization</td></tr>
<tr><td><b>Location</b></td><td></td></tr>
<tr><td><b>Description</b></td><td>Unset root SUID after calling chroot().<br>The chroot() function establishes a virtual root directory for the owning process. This may be used to limit the amount of file system access a potential hacker could use if he or she gained control of the process. Programs like ftp and httpd commonly make use of this function.<br>The chroot() function requires root (superuser) access to call. If the programmer continues to run as root after the chroot() call, he or she opens up a potential vulnerability window for an attacker to use elevated privilege.<br>Use of chroot is desirable but should also be a flag to indicate that one needs to carefully check to ensure that related security issues are addressed.</td></tr>
<tr><td><b>APIs</b></td><td><table><tr><th>FunctionName</th><th>Comments</th></tr><tr><td>chroot</td><td></td></tr></table></td></tr>
<tr><td><b>Method of Attack</b></td><td>The method of attack depends on what other security holes are present that a hacker can exploit. This problem does not create the security hole per se but increases the damage a hacker can do after exploiting a hole to gain control of the process.</td></tr>
<tr><td><b>Exception Criteria</b></td><td></td></tr>
<tr><td><b>Solutions</b></td><td><table><tr><th>Solution Applicability</th><th>Solution Description</th><th>Solution Efficacy</th></tr><tr><td>Whenever chroot is used.</td><td>ASAP following the chroot() call,</td><td></td></tr></table></td></tr>
</table>

[1] http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

| | |
|---|---|
| | programmer should set the EUID and UID to a less-privileged user. |
| **Signature Details** | int chroot(const char *) |
| **Examples of Incorrect Code** | `[...]`<br>`char path[] = "/usr/sandbox";`<br>`chroot(path);`<br>`[...]`<br>`/* Continuing without changing`<br>`user ID is a security risk because`<br>`running as root. */` |
| **Examples of Corrected Code** | `[...]`<br>`char path[] = "/usr/sandbox";`<br>`close(anOpenFile); /* Should not`<br>`leave file descriptors open. */`<br>`if (chroot(path)) exit(1); /*`<br>`Should check return value. */`<br>`chdir("/"); /* Must do this`<br>`or chroot() won't have intended`<br>`effect */`<br>`setegid(ogid); /* Should change`<br>`group ID */`<br>`seteuid(ouid); /* Should change`<br>`user ID */`<br>`[...]`<br>`/* Now can safely continue */` |
| **Source Reference** | • Viega, John & McGraw, Gary. Building Secure Software: How to Avoid Security Problems the Right Way. Boston, MA: Addison-Wesley Professional, 2001, ISBN: 020172152X, pg. 204. |
| **Recommended Resources** | • chroot man page[2]<br>• Bishop, Matt & Dilger, Michael. Checking for Race Conditions in File Accesses[3], 1996 |
| **Discriminant Set** | **Operating System**: • UNIX (All)<br>**Languages**: • C<br>• C++ |

## Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

[1] mailto:copyright@cigital.com